Exhibit 2

Method Claim: 1

| US11080885	Zapworks Designer ("The accused instrumentality")
1. A method of providing an augmented reality experience with a user device, comprising:	The accused instrumentality practices a method of providing an augmented reality experience with a user device (e.g., a user device such as a smartphone). As shown below, Zapworks Designer provides an augmented reality experience via a mobile web browser by overlaying virtual elements on top of the real-world view. https://zap.works/designer/

The easiest way to create immersive WebAR

Our drag and drop editor enables you to effortlessly combine image, world, and face tracked scenes to produce immersive AR experiences.
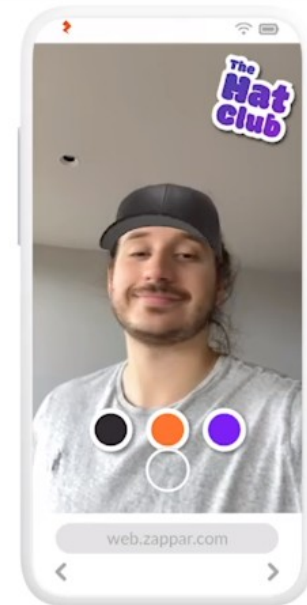
https://zap.works/designer/
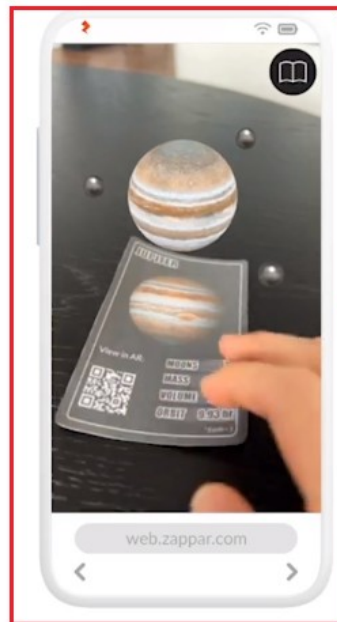
Image tracking   World tracking   Face tracking

https://zap.works/designer/
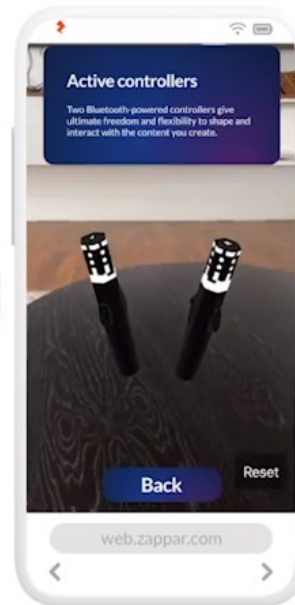
Alpha channel video

Alpha channel is an additional channel that defines areas of transparency. Add videos with alpha channel to any Designer project.

WebAR

WebAR allows users to view experiences using their browser. It is a great way to ensure end users can access content with no friction.

Self hosting

Available on the enterprise plan, Self-hosting allows customisation while also being a secure way to host and serve WebAR experiences.
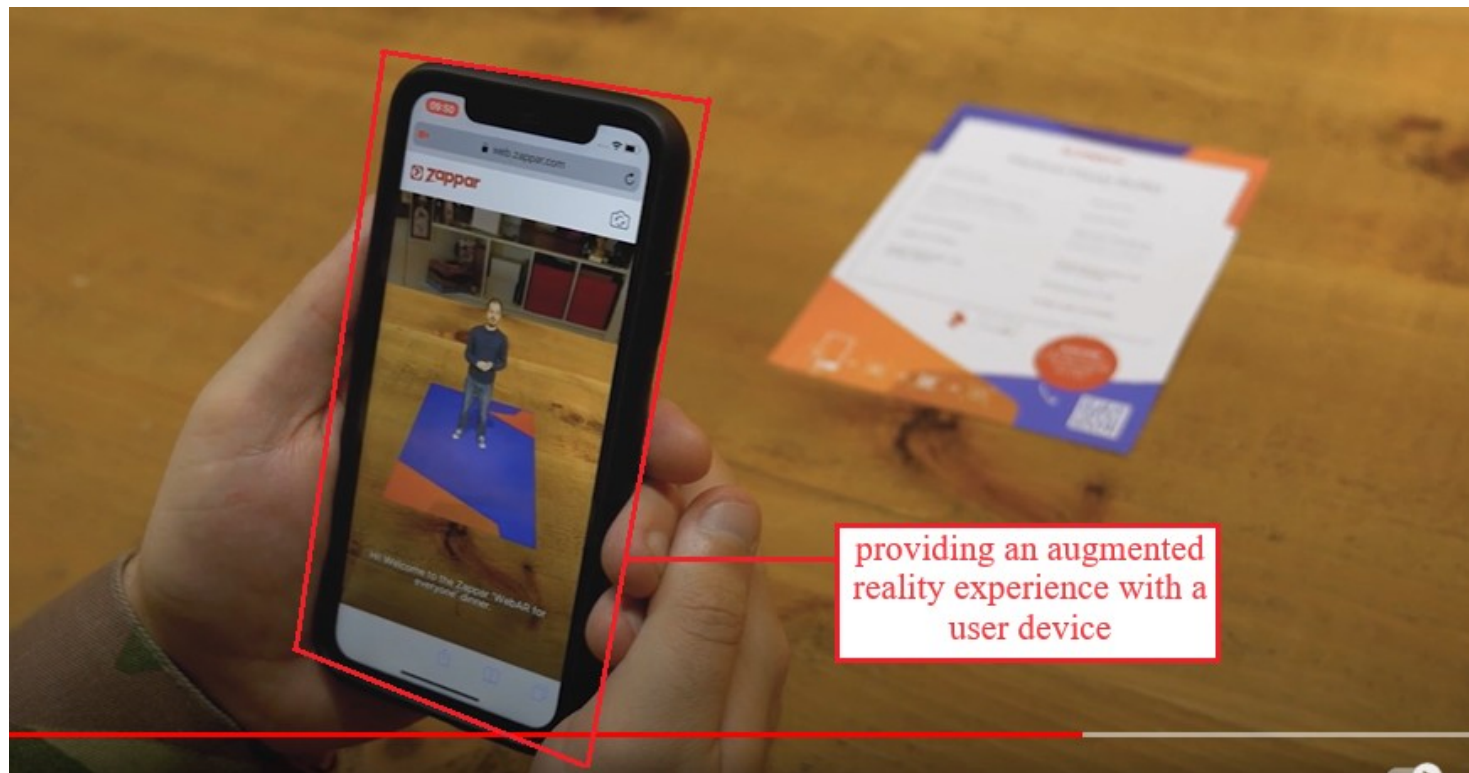
https://zap.works/designer/

Getting Started

Designer

Zapworks Designer is a browser-based, no-code Augmented Reality development tool. Designer gives you the platform to create immersive World, Image and Face tracked augmented reality in just a few clicks. Whether you're looking to bring your print to life or add interaction to your static content, it can all be done with our easy-to-use tool.

https://docs.zap.works/designer/?_ga=2.233641118.1470678592.1705033547-2123639248.1705033547

WebAR

AR projects built using the Zapworks suite of tools can be run directly in the default mobile browser, this is known as WebAR. While the Zappar platform and Universal AR SDKs work well on the browsers that enjoy the vast majority of mobile market-share, there are a number of web browsers available across the mobile and desktop device landscape which may have some caveats.
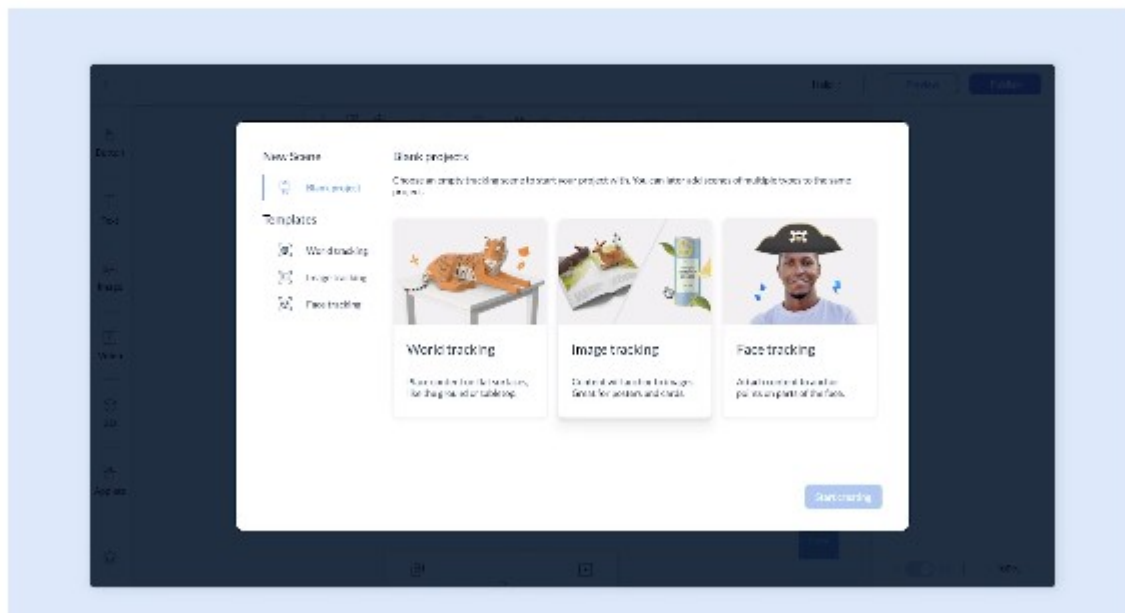
https://docs.zap.works/general/platform/supported-devices/

providing an augmented reality experience with a user device

	https://www.youtube.com/watch?v=JxEGUZWZ7d8
receiving an input image of a physical environment from a camera included in or on the user device, wherein a digitally encoded marker (DEM) is positioned at a marker location within the physical environment;	The accused instrumentality discloses receiving an input image of a physical environment (e.g., an input image of the target image with a printed QR code placed in a physical environment), from a camera (e.g., a camera of the device used for scanning QR code) included in or on the user device (e.g., a user device such as a smartphone), wherein a digitally encoded marker (DEM) (e.g., a target image with a printed QR code for generating WebAR experience) is positioned at a marker location within the physical environment (e.g., a location of the target image placed within the environment). As shown below, Zapworks Designer uses a QR code to access a website for experiencing augmented reality. An image of a target image with a printed QR code is captured by the camera of the smartphone device. The physically printed target image is positioned within the physical environment. **Image Tracking** Designer Image Tracking Tracking **Image Tracking** can detect and track digital content to a flat or upright image. This is great for building content that's augmented onto print material such as business cards, posters, and magazine pages. https://docs.zap.works/designer/tracking/image-tracking/

Upon selecting **Image Tracking**, you'll be prompted to upload a **target image** in which your content will track to. To do this, select the **Upload target image** button in the **Project Properties** panel. Your target image will be trained, and once uploaded, will appear on your canvas.

https://docs.zap.works/designer/tracking/image-tracking/

To track AR content to objects such as posters, packaging, birthday cards or magazines, we use Image Tracking. Image Tracking uses images as targets to track AR content. These images are known as either **Target Images** or **Tracking Images.**

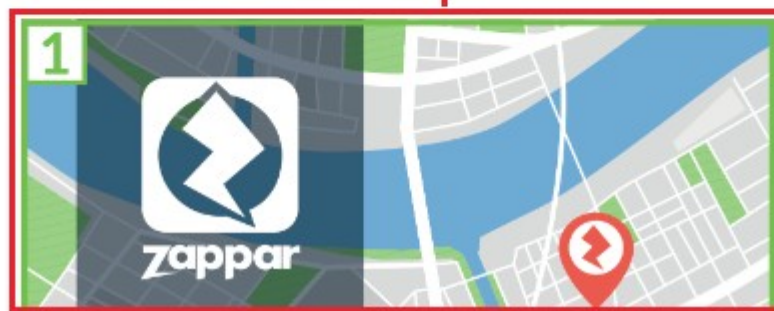Image Tracking has a few benefits over the other tracking types:

- Reliable stability and tracking performance using Zappar's best-in-class image tracking technology.

- Add interactivity to physical objects, static objects, and print material.

https://docs.zap.works/general/design/what-makes-good-tracking-image/

What makes up a Tracking Image?

An effective tracking image usually incorporates three components:

1. An Image — digitally encoded marker (DEM)

2. A **Call To Action** (CTA) - used to tell the user how to activate AR content.

3. A **Trigger** (zapcode / QR code) - used to launch the AR content once scanned.

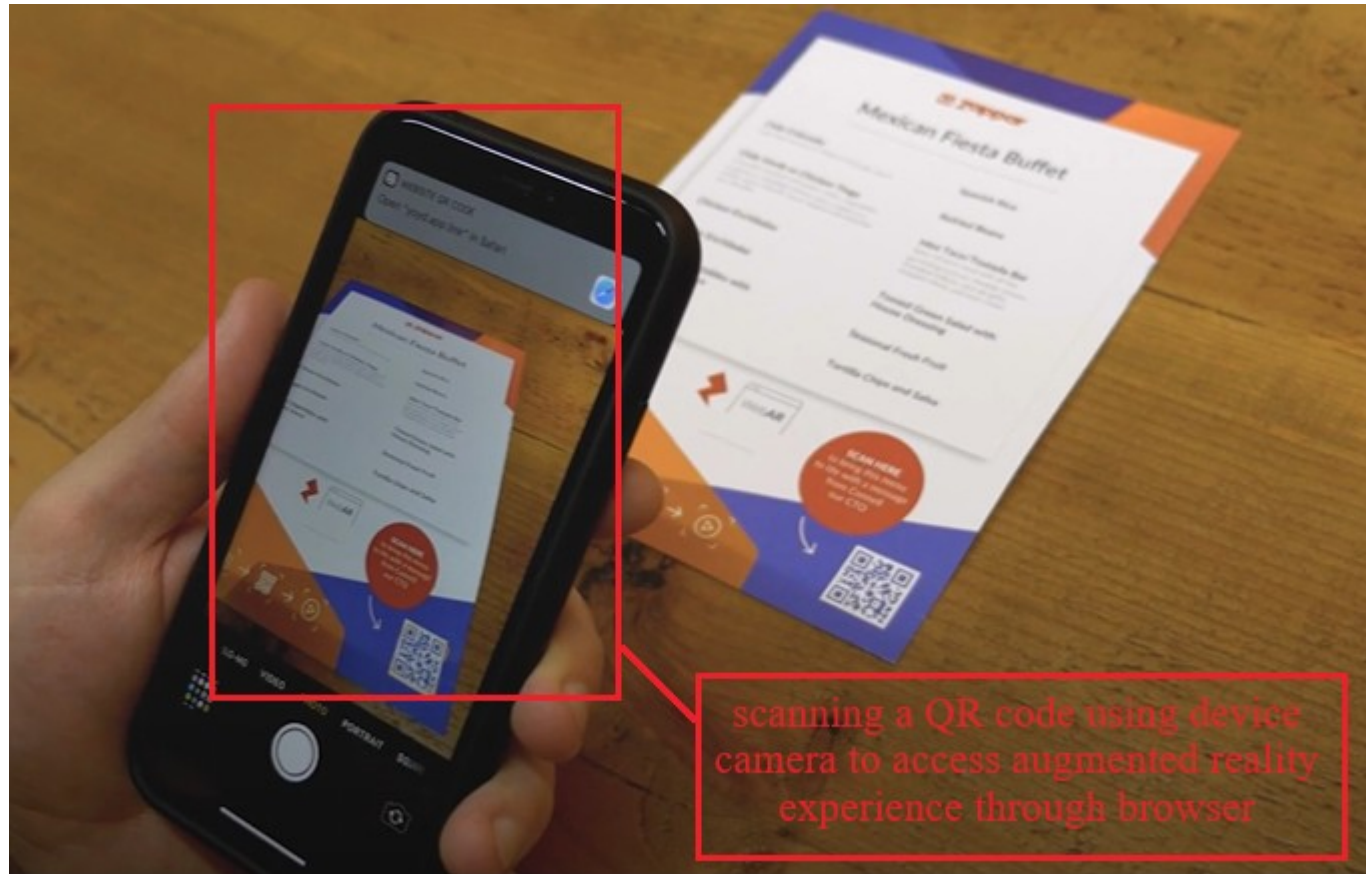https://docs.zap.works/general/design/what-makes-good-tracking-image/

Previewing

When you **Preview** your experience, you will be provided with a test QR code. Scanning the test QR code using a QR code scanner (built in to most native camera applications) will launch a preview of your experience. This is a reflection of what users will see when they scan your QR code trigger.

Image-Tracking

Clicking the **Preview** button will open a **Preview Modal** where a test QR code and the target image you uploaded will be displayed.

https://docs.zap.works/designer/previewing-and-publishing/

https://www.youtube.com/watch?v=JxEGUZWZ7d8
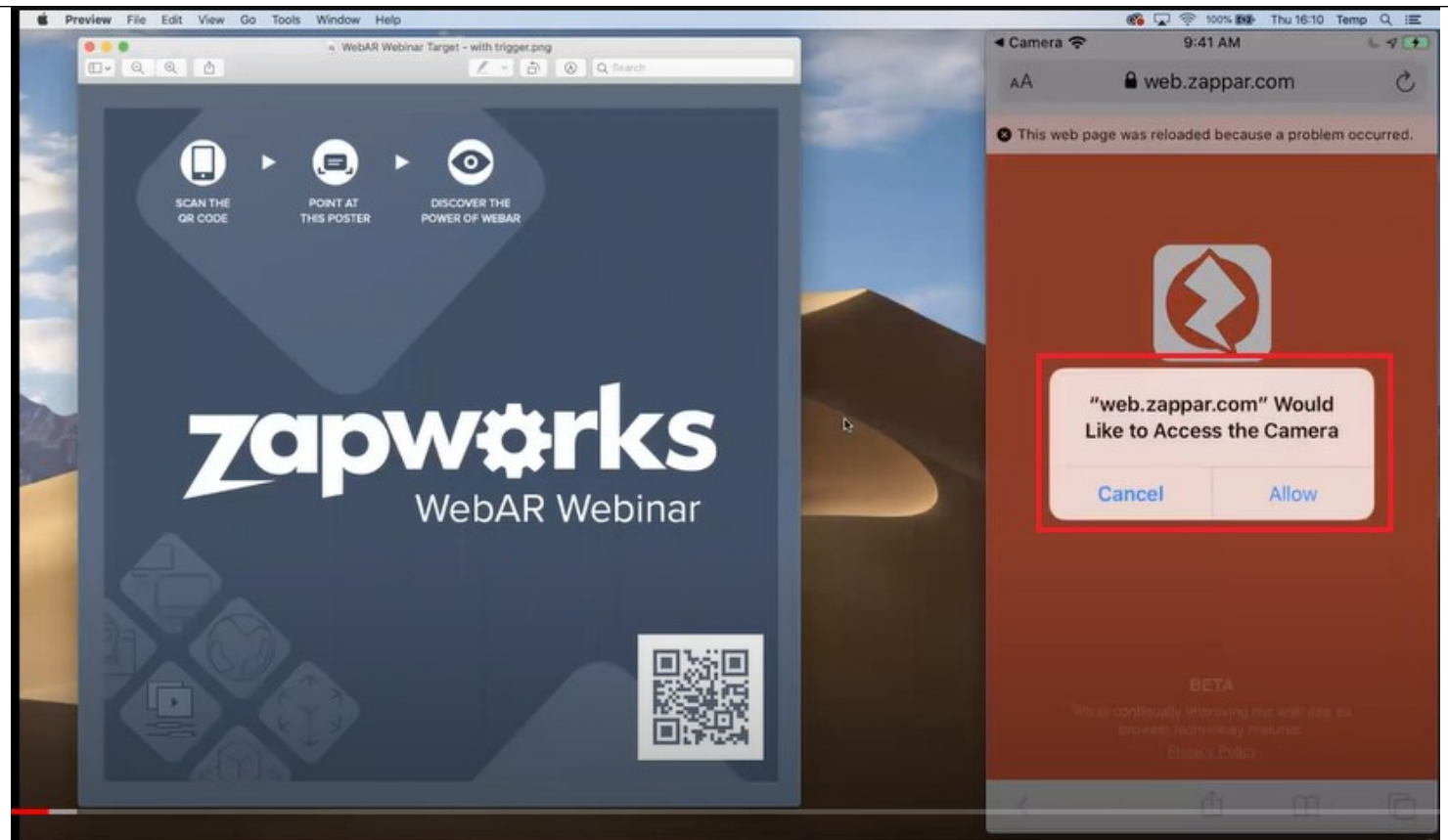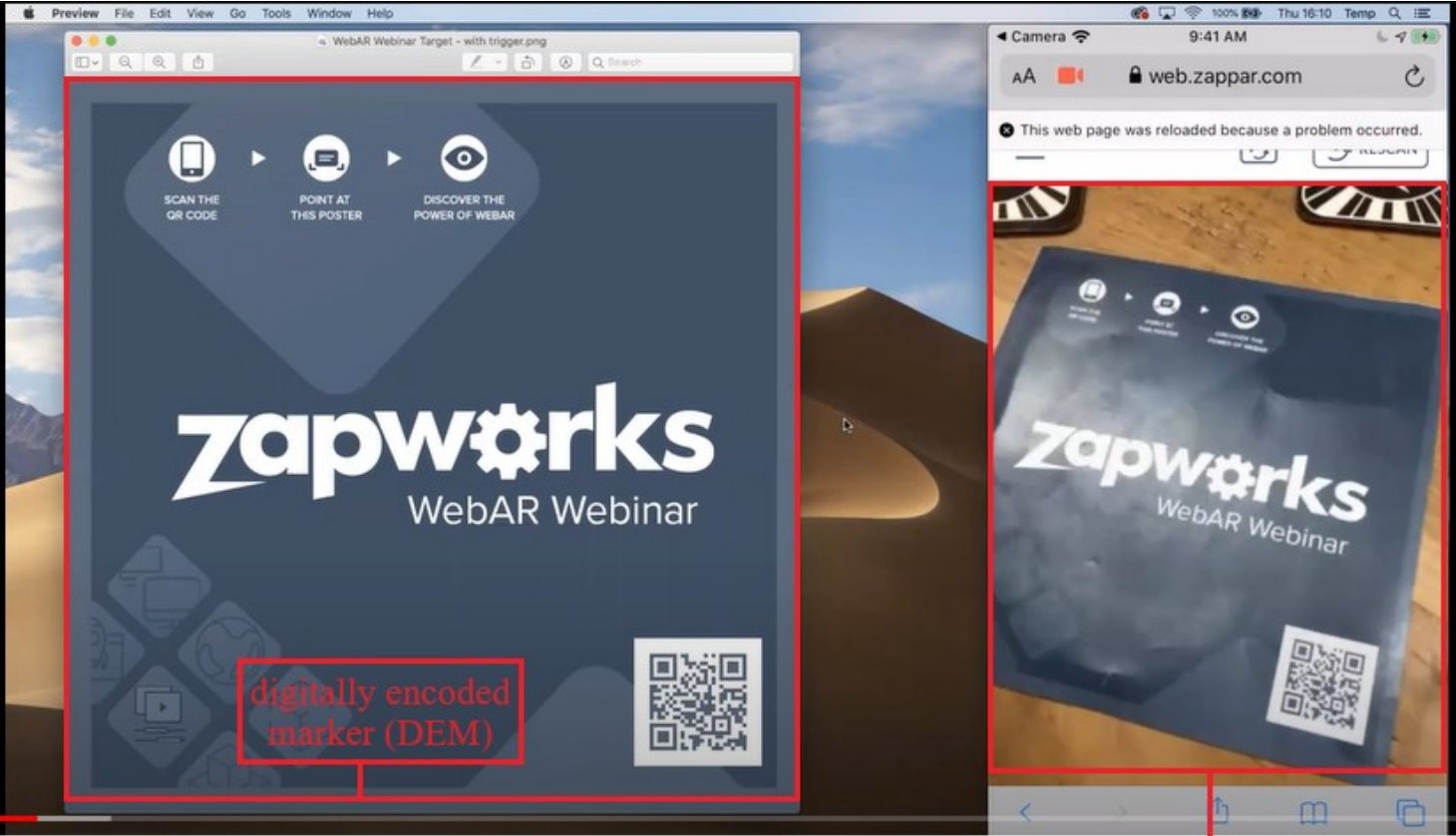
https://www.youtube.com/watch?v=JxEGUZWZ7d8

https://www.youtube.com/watch?v=iWKZ2DF3v_c

receiving an input image of a physical environment from a camera included in or on the user device, wherein a digitally encoded marker (DEM) is positioned at a marker location within the physical environment

https://www.youtube.com/watch?v=iWKZ2DF3v_c

| decoding data from the DEM by processing | The accused instrumentality practices decoding data (e.g., anchor pose data relative to world origin) from the DEM (e.g., a target image with a printed QR code for generating WebAR experience) by processing the input image (e.g., an input image of the physical |

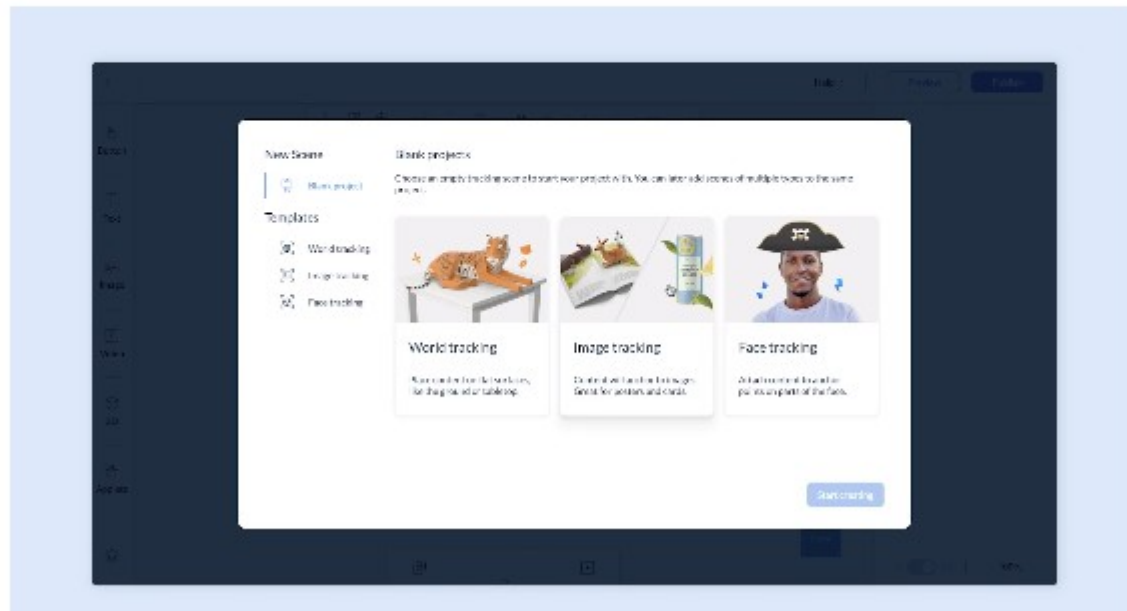|	
the input image, wherein the decoded data comprises at least one of geographic coordinate data and relative coordinate data;	environment captured by the camera of the device used for scanning QR code), wherein the decoded data (e.g., anchor pose data relative to world origin) comprises at least one of geographic coordinate data and relative coordinate data (e.g., position data of an Anchor pose relative to world origin). As shown below, the target image is tracked in the video obtained from the camera of the user device. An anchor pose of the Image Anchor type is used to obtain the position of the target image relative to the world origin. **Image Tracking** (Designer) (Image Tracking) (Tracking) **Image Tracking** can detect and track digital content to a flat or upright image. This is great for building content that's augmented onto print material such as business cards, posters, and magazine pages. https://docs.zap.works/designer/tracking/image-tracking/

Upon selecting **Image Tracking**, you'll be prompted to upload a **target image** in which your content will track to. To do this, select the **Upload target image** button in the **Project Properties** panel. Your target image will be trained, and once uploaded, will appear on your canvas.

https://docs.zap.works/designer/tracking/image-tracking/

To track AR content to objects such as posters, packaging, birthday cards or magazines, we use Image Tracking. Image Tracking uses images as targets to track AR content. These images are known as either **Target Images** or **Tracking Images.**

Image Tracking has a few benefits over the other tracking types:

- Reliable stability and tracking performance using Zappar's best-in-class image tracking technology.

- Add interactivity to physical objects, static objects, and print material.

https://docs.zap.works/general/design/what-makes-good-tracking-image/

What makes up a Tracking Image?

An effective tracking image usually incorporates three components:

1. An Image — digitally encoded marker (DEM)

2. A **Call To Action** (CTA) - used to tell the user how to activate AR content.

3. A **Trigger** (zapcode / QR code) - used to launch the AR content once scanned.

https://docs.zap.works/general/design/what-makes-good-tracking-image/

Previewing

When you **Preview** your experience, you will be provided with a test QR code. Scanning the test QR code using a QR code scanner (built in to most native camera applications) will launch a preview of your experience. This is a reflection of what users will see when they scan your QR code trigger.

Image-Tracking

Clicking the **Preview** button will open a **Preview Modal** where a test QR code and the target image you uploaded will be displayed.

https://docs.zap.works/designer/previewing-and-publishing/

Coordinate Systems and Poses

(Getting Started)   (JavaScript)   (Universal AR)

The Zappar library models the 3D space of an AR experience using three transformations:

Transformation	Description
Projection matrix	Applies a perspective projection corresponding to a camera with a focal length, as well as near and far clipping planes
Camera pose	Applies the position and rotation of the camera relative to a 'world' origin
Anchor pose	Applies the position, scale and rotation of a given tracked object (e.g. an image, or face) relative to the world origin

decoding data

https://docs.zap.works/universal-ar/javascript/getting-started/coordinate-systems-and-poses/

Anchor Pose

Each of the tracking types provided by the Zappar library expose anchors with a function to obtain an anchor pose for a given camera pose, e.g.:

The following sections gives more details about the various tracking types and their associated anchors:

- Image Tracking

- Instant World Tracking

- Face Tracking

https://docs.zap.works/universal-ar/javascript/getting-started/coordinate-systems-and-poses/

Image Anchors

Each `ImageTracker` exposes anchors for images detected and tracked in the camera view. At this time, `ImageTracker`s only track one image in view at a time.

Anchors have the following parameters:

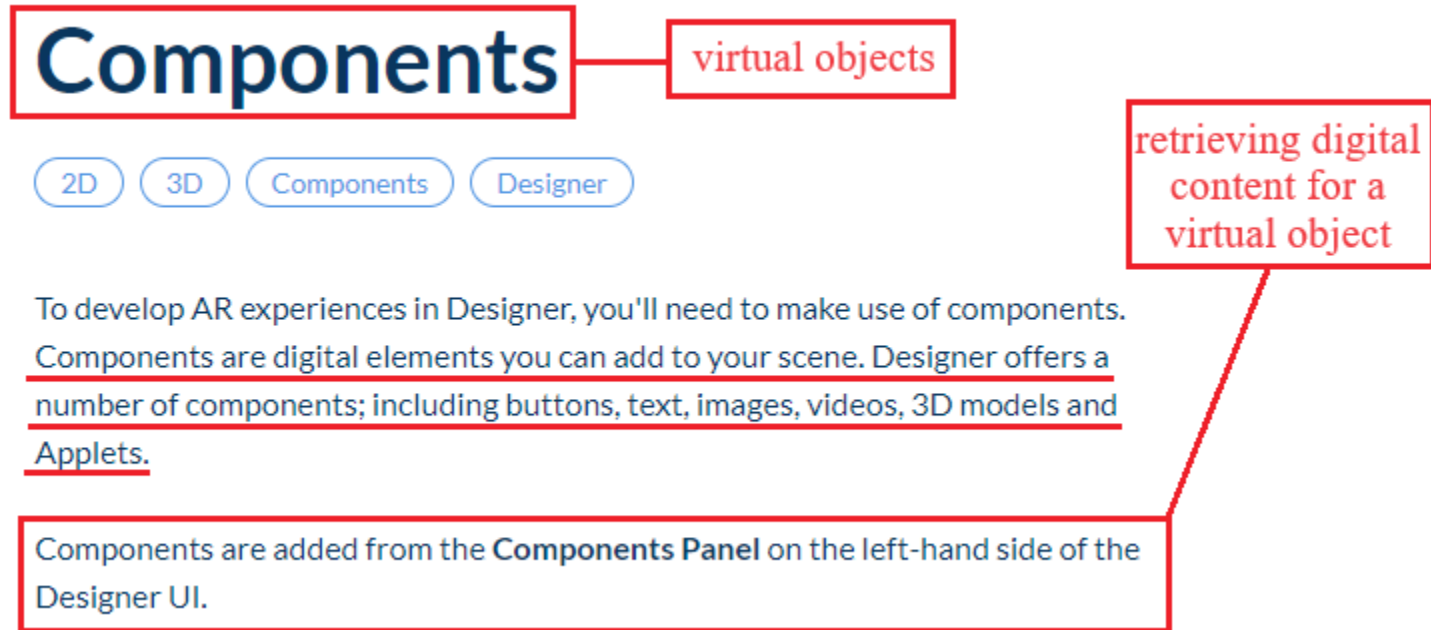decoded data comprises relative coordinate data

Parameter	Description
`pose(cameraPose: Float32Array, mirror?: boolean)`	A function that returns the pose matrix for this anchor
`poseCameraRelative(mirror?: boolean)`	A function that returns the pose matrix (relative to the camera) for this anchor, for use with `cameraPoseWithOrigin`

https://docs.zap.works/universal-ar/javascript/tracking/image-tracking/

| retrieving digital content for a virtual object; and	The accused instrumentality practices retrieving digital content (e.g., digital content such as digital files used for adding components such as text, 3D models, etc. to the AR experience) for a virtual object (e.g., Components that are digital elements to add to an AR experience such as buttons, text, images, videos, 3D models and Applets).

As shown below, digital contents for a virtual object such as text, 3D models, etc. are retrieved and augmented over a target image.

Components — virtual objects

2D   3D   Components   Designer

retrieving digital content for a virtual object

To develop AR experiences in Designer, you'll need to make use of components. Components are digital elements you can add to your scene. Designer offers a number of components; including buttons, text, images, videos, 3D models and Applets.

Components are added from the **Components Panel** on the left-hand side of the Designer UI.

https://docs.zap.works/designer/components/

Components are added from the **Components Panel** on the left-hand side of the Designer UI.

for retrieving digital content for a virtual object

Components Panel

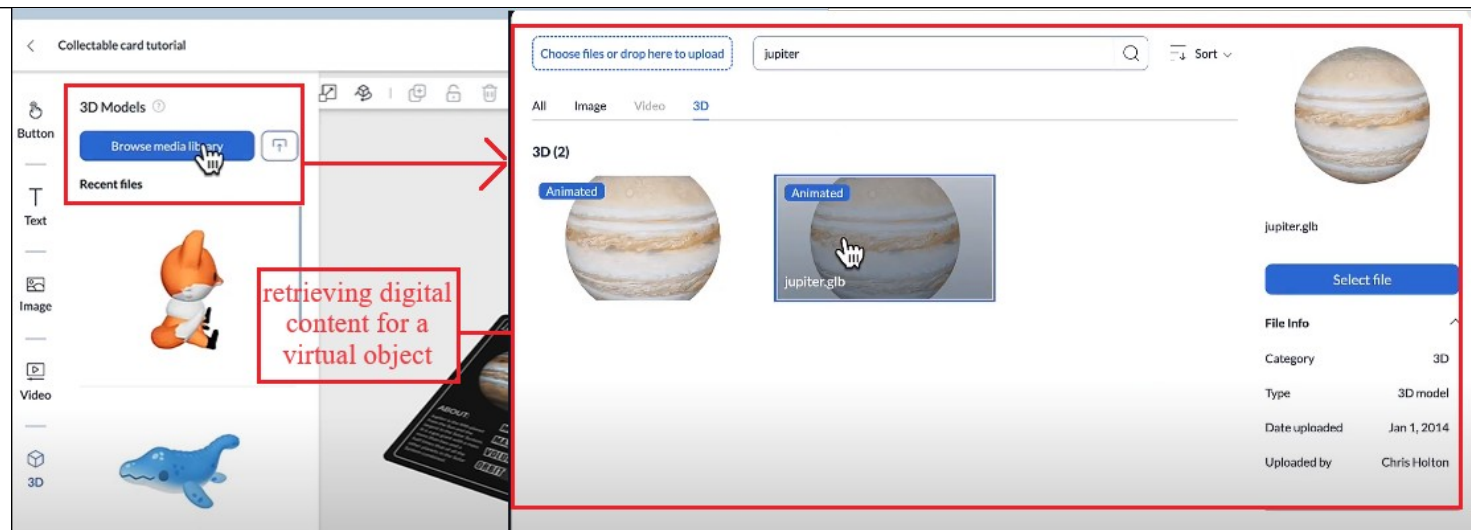https://docs.zap.works/designer/components/
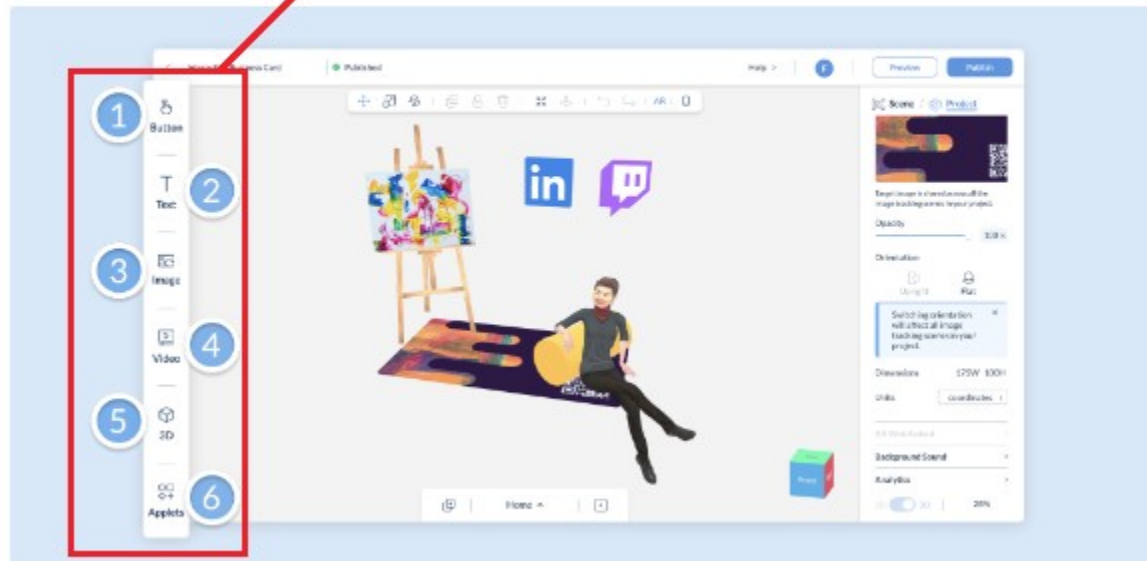
This articles in this section showcase these different components, explain how they can be added and how you can edit their properties.

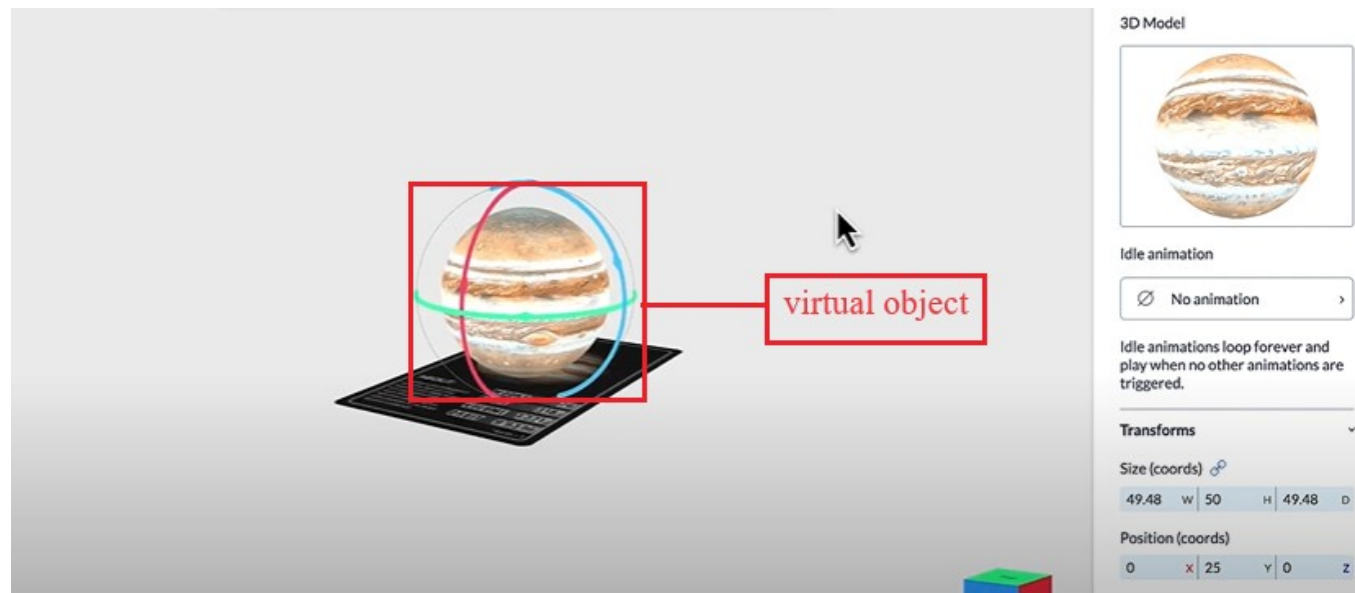virtual objects

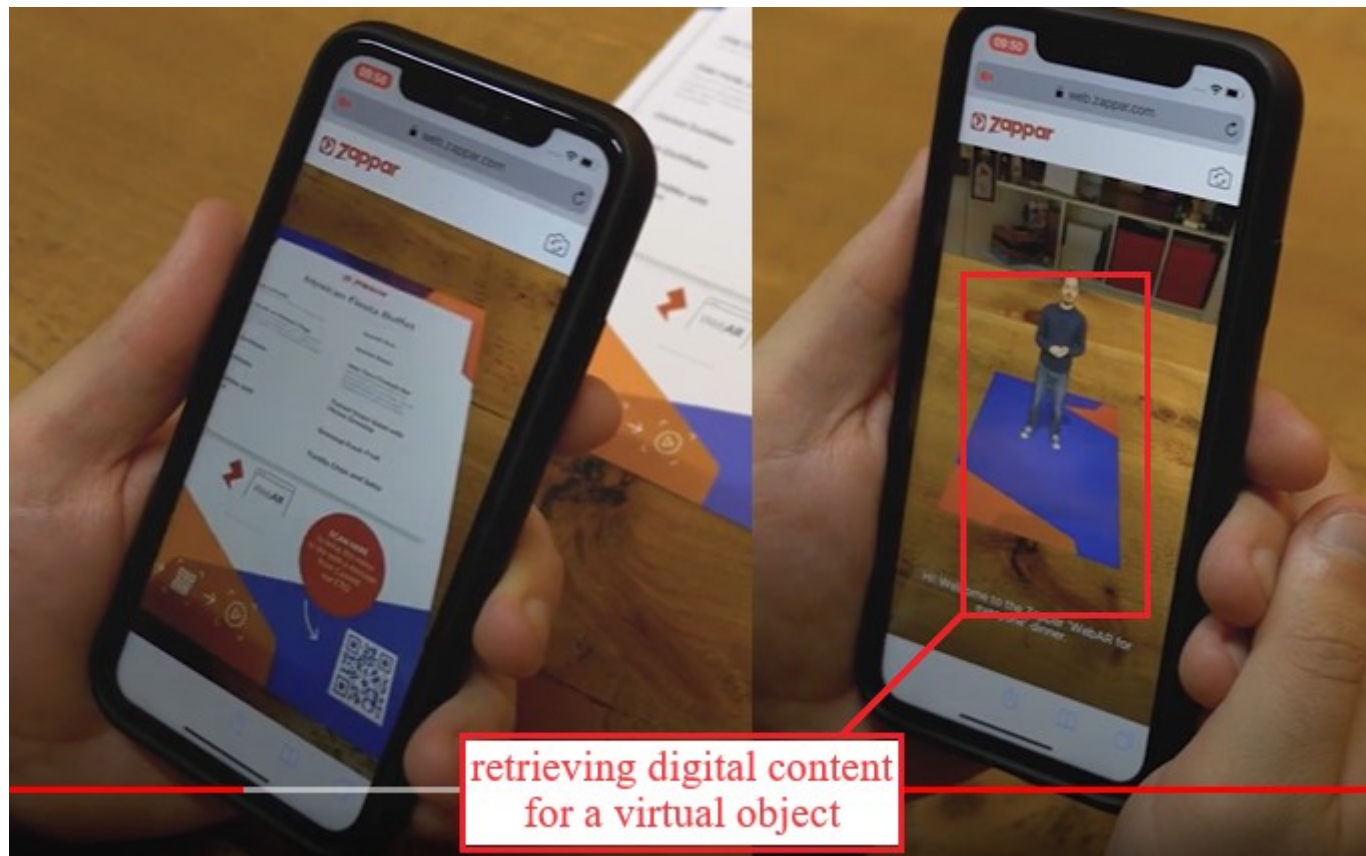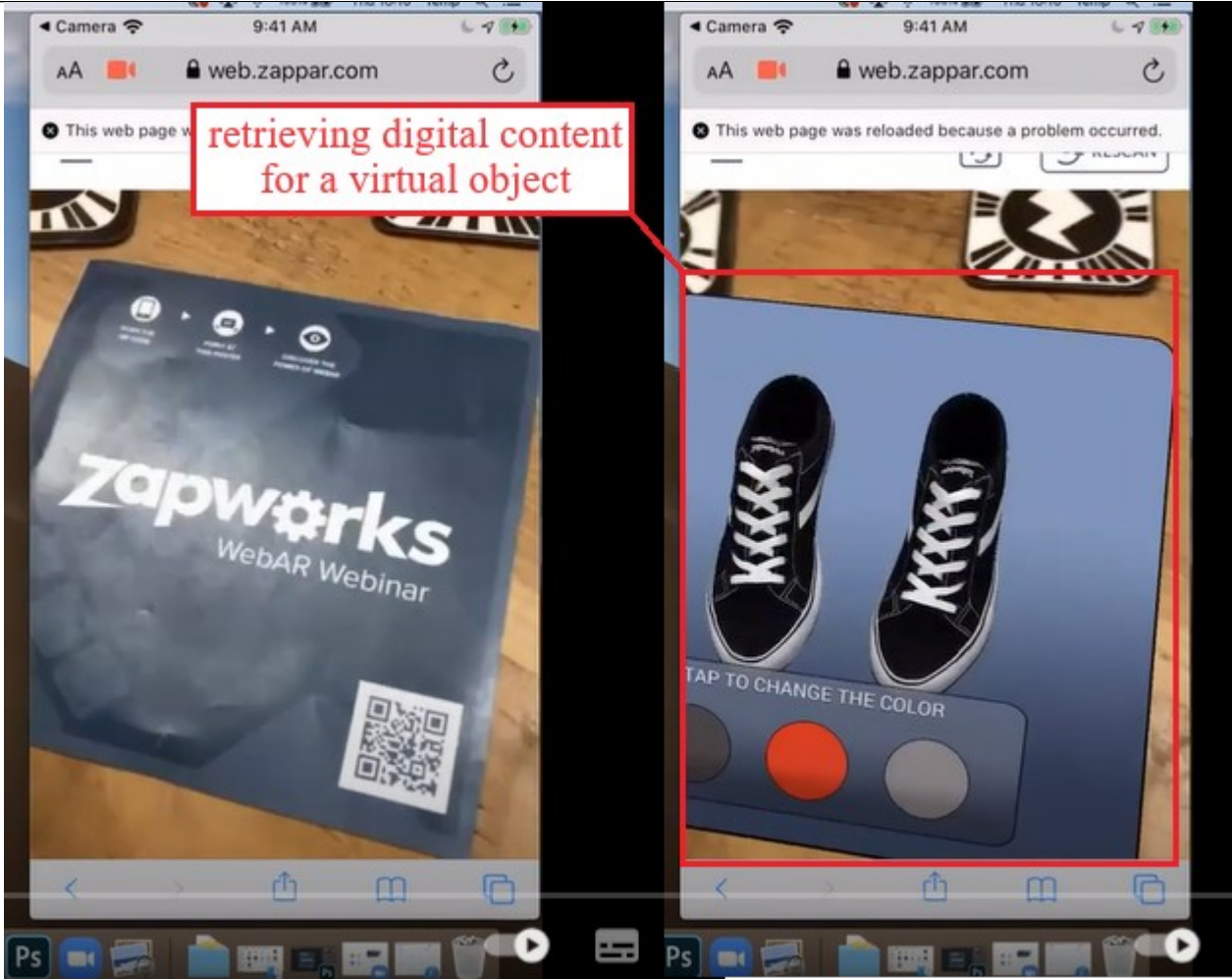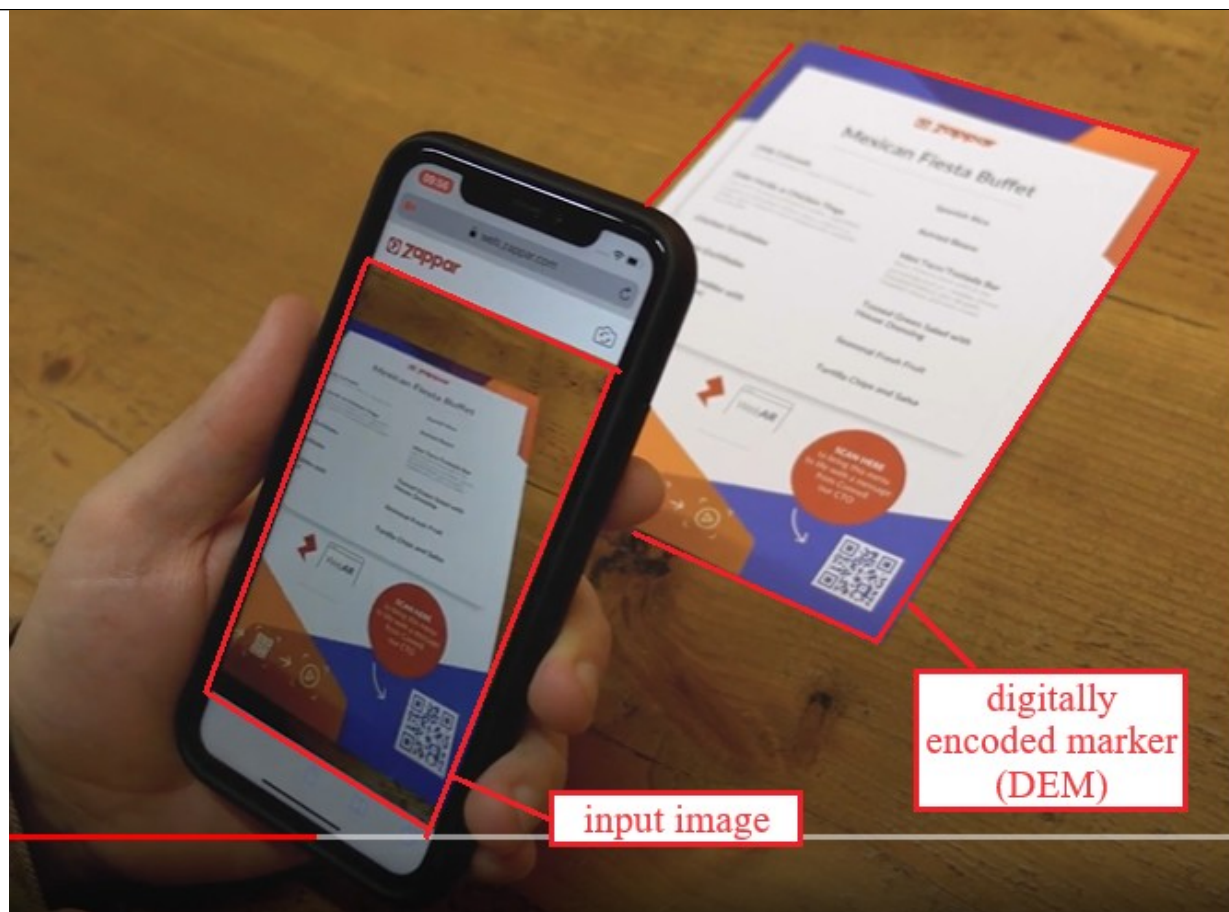| Component	Description
(1) Button	Select from a range of basic buttons to add to your scene.
(2) Text	Choose from preset themed and hierarchy based text objects.
(3) Image	Add an image from your media library to your scene, or upload a new image from your device.
(4) Video	Add a video from your media library to your scene, or upload a new video from your device.
(5) 3D	Add a 3D model to your scene to create a three-dimensional experience.
(6) Applets	Create more elaborate functionalities quickly and easily.

https://docs.zap.works/designer/components/

https://www.youtube.com/watch?v=FY1bDxyl2Aw&t=225s

https://www.youtube.com/watch?v=FY1bDxyl2Aw&t=225s

retrieving digital content
for a virtual object

https://www.youtube.com/watch?v=JxEGUZWZ7d8

https://www.youtube.com/watch?v=iWKZ2DF3v_c

| displaying an augmented reality image, on a display | The accused instrumentality discloses displaying an augmented reality image (e.g., an AR image with an overlay of a virtual object), on a display screen (e.g., a display screen of the device used for scanning QR code) of the user device (e.g., a user device such as a smartphone), including the input image (e.g., an input image of the physical environment |

| screen of the user device, including the input image and an overlay image representing the virtual object positioned within the augmented reality image based on the decoded data from the DEM and the marker location, | captured by the camera of the device used for scanning QR code) and an overlay image (e.g., an overlay image of the virtual object) representing the virtual object (e.g., Components that are digital elements to add to an AR experience such as buttons, text, images, videos, 3D models and Applets) positioned within the augmented reality image (e.g., an AR image with an overlay of a virtual object) based on the decoded data (e.g., anchor pose data relative to world origin) from the DEM (e.g., a target image with a printed QR code for generating WebAR experience) and the marker location (e.g., a location of the target image placed within the environment).

As shown below, an augmented reality image is displayed on the display of the device used for scanning the QR code. The augmented image contains the camera feed as well as the virtual object which is overlayed at a position relative to the center of the target image. The Image Anchor is used to obtain the position of the target image relative to the world space. The center of the image is used as the origin for positioning the virtual objects. |

https://www.youtube.com/watch?v=JxEGUZWZ7d8

virtual object

displaying an augmented reality image, on a display screen of the user device, including the input image and an overlay image representing the virtual object positioned within the augmented reality

https://www.youtube.com/watch?v=JxEGUZWZ7d8

https://www.youtube.com/watch?v=iWKZ2DF3v_c

virtual object

displaying an augmented reality image, on a display screen of the user device, including the input image and an overlay image representing the virtual object positioned within the augmented reality

https://www.youtube.com/watch?v=iWKZ2DF3v_c

overlay image representing the virtual object

https://www.youtube.com/watch?v=FY1bDxyI2Aw&t=225s

Coordinate Systems and Poses

(Getting Started)   (JavaScript)   (Universal AR)

The Zappar library models the 3D space of an AR experience using three transformations:

Transformation	Description
Projection matrix	Applies a perspective projection corresponding to a camera with a focal length, as well as near and far clipping planes
Camera pose	Applies the position and rotation of the camera relative to a 'world' origin
Anchor pose	Applies the position, scale and rotation of a given tracked object (e.g. an image, or face) relative to the world origin

decoding data

https://docs.zap.works/universal-ar/javascript/getting-started/coordinate-systems-and-poses/

Anchor Pose

Each of the tracking types provided by the Zappar library expose anchors with a function to obtain an anchor pose for a given camera pose, e.g.:

The following sections gives more details about the various tracking types and their associated anchors:

- Image Tracking

- Instant World Tracking

- Face Tracking

https://docs.zap.works/universal-ar/javascript/getting-started/coordinate-systems-and-poses/

Image Anchors

Each `ImageTracker` exposes anchors for images detected and tracked in the camera view. At this time, `ImageTracker` s only track one image in view at a time.

Anchors have the following parameters:

decoded data comprises relative coordinate data

Parameter	Description
`pose(cameraPose: Float32Array, mirror?: boolean)`	A function that returns the pose matrix for this anchor
`poseCameraRelative(mirror?: boolean)`	A function that returns the pose matrix (relative to the camera) for this anchor, for use with `cameraPoseWithOrigin`

https://docs.zap.works/universal-ar/javascript/tracking/image-tracking/

Image Tracking

(A-Frame) (Tracking) (Universal AR)

Image tracking can detect and track a flat image in 3D space. This is great for building content that's augmented onto business cards, posters, magazine pages, etc.

The group provides a coordinate system that has its origin at the center of the image, with positive X axis to the right, the positive Y axis towards the top and the positive Z axis coming up out of the plane of the image. The scale of the coordinate system is such that a Y value of +1 corresponds to the top of the image, and a Y value of -1 corresponds to the bottom of the image. The X axis positions of the left and right edges of the target image therefore depend on the aspect ratio of the image.
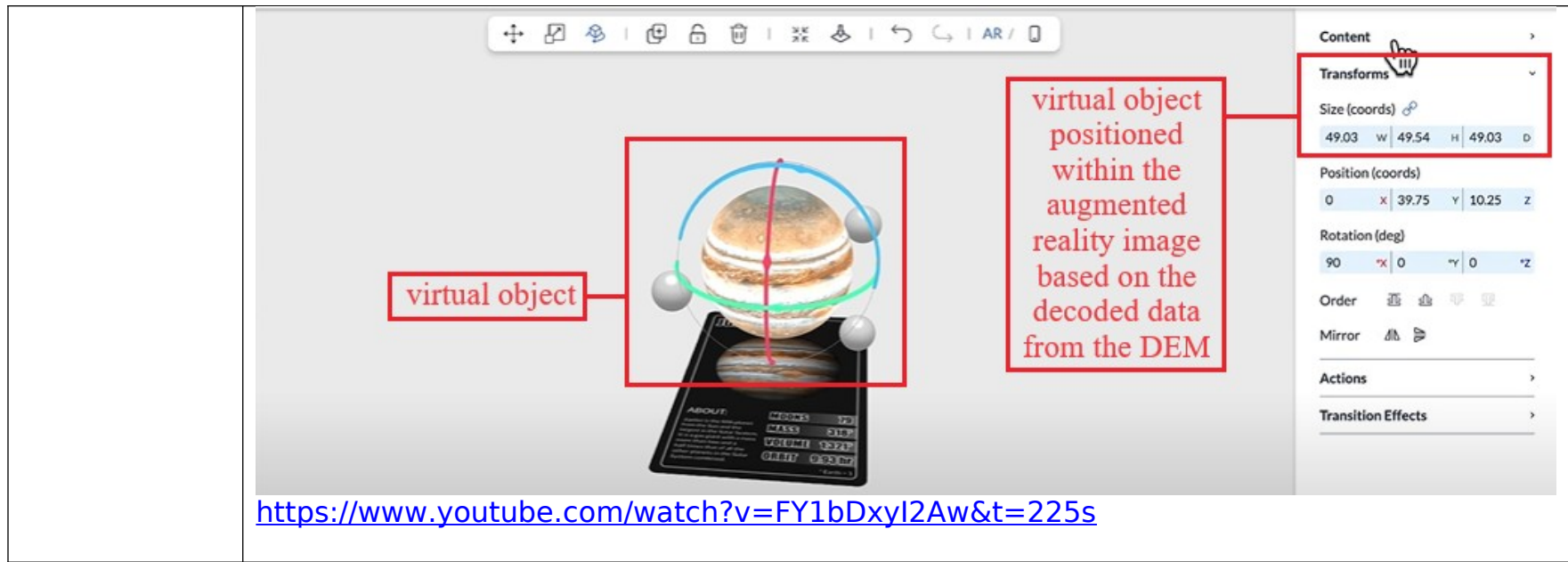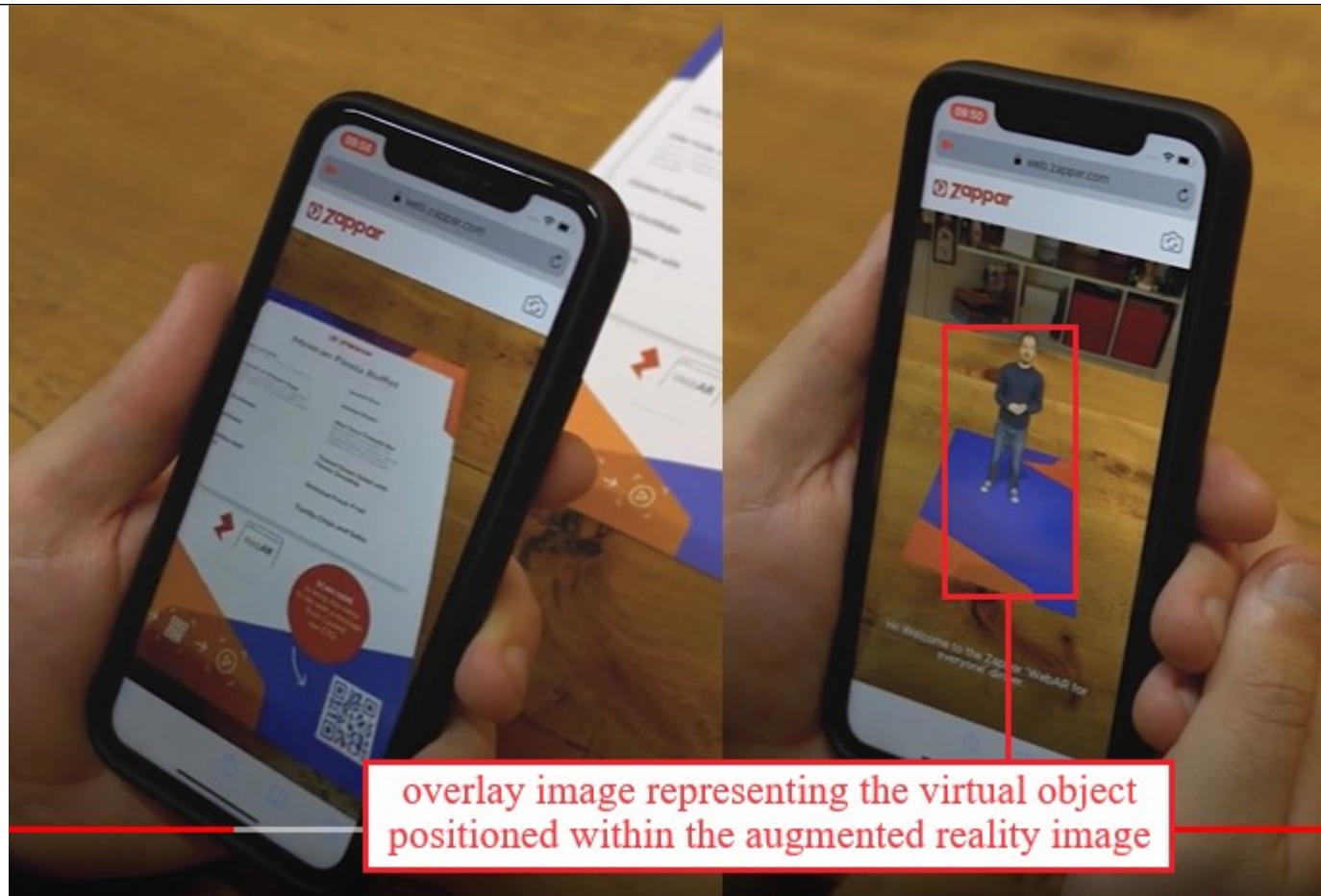
https://docs.zap.works/universal-ar/a-frame/tracking/image-tracking/

You may use the **Properties Panel** to make the following adjustments to your 3D model:

Property	Description
Content	Replace the 3D component with another 3D component you have uploaded to the Media Library.
Transforms	Edit the **Size** (Width, Height and Depth), **Position (X/Y/Z coordinates)**, and **Rotation** (degrees on the X/Y/Z axes) of the 3D model. You can also **Mirror** the 3D component and change the 3D model's layer **Order** (forwards or backwards) relative to other scene components.
Actions	Create interactivity by adding an on tap **Action** to the 3D component.
Transition Effects	Add an **Enter** or **Exit** transition effect to the 3D component.

virtual object positioned within the augmented reality image based on the decoded data from the DEM and the marker location

https://docs.zap.works/designer/3d/using-3d/

https://www.youtube.com/watch?v=FY1bDxyI2Aw&t=225s

overlay image representing the virtual object positioned within the augmented reality image

https://www.youtube.com/watch?v=JxEGUZWZ7d8

overlay image representing the virtual object positioned within the augmented reality image

https://www.youtube.com/watch?v=iWKZ2DF3v_c

wherein the overlay image is positioned within the	The accused instrumentality discloses wherein the overlay image (e.g., an overlay image of the virtual object) is positioned within the augmented reality image (e.g., an AR image with an overlay of a virtual object) using the at least one of the geographic coordinate data and the relative coordinate data (e.g., position data of an Anchor pose relative to

| augmented reality image using the at least one of the geographic coordinate data and the relative coordinate data decoded from the DEM. | world origin) decoded from the DEM (e.g., a target image with a printed QR code for generating WebAR experience).

As shown below, the augmented image contains the camera feed as well as the virtual object which is overlayed at a position relative to the center of the target image. The Image Anchor is used to obtain the position of the target image relative to the world space. The center of the image is used as the origin for positioning the virtual objects. The position of the virtual object can be adjusted by the user with respect to the center of the target image. |

Coordinate Systems and Poses

Getting Started   JavaScript   Universal AR

The Zappar library models the 3D space of an AR experience using three transformations:

Transformation	Description
Projection matrix	Applies a perspective projection corresponding to a camera with a focal length, as well as near and far clipping planes
Camera pose	Applies the position and rotation of the camera relative to a 'world' origin
Anchor pose	Applies the position, scale and rotation of a given tracked object (e.g. an image, or face) relative to the world origin

decoding data

https://docs.zap.works/universal-ar/javascript/getting-started/coordinate-systems-and-poses/

Anchor Pose

Each of the tracking types provided by the Zappar library expose anchors with a function to obtain an anchor pose for a given camera pose, e.g.:

The following sections gives more details about the various tracking types and their associated anchors:

- **Image Tracking**

- **Instant World Tracking**

- **Face Tracking**

https://docs.zap.works/universal-ar/javascript/getting-started/coordinate-systems-and-poses/

Image Anchors

Each `ImageTracker` exposes anchors for images detected and tracked in the camera view. At this time, `ImageTracker`s only track one image in view at a time.

Anchors have the following parameters:

decoded data comprises relative coordinate data

Parameter	Description
`pose(cameraPose: Float32Array, mirror?: boolean)`	A function that returns the pose matrix for this anchor
`poseCameraRelative(mirror?: boolean)`	A function that returns the pose matrix (relative to the camera) for this anchor, for use with `cameraPoseWithOrigin`

https://docs.zap.works/universal-ar/javascript/tracking/image-tracking/

Image Tracking

(A-Frame)   (Tracking)   (Universal AR)

Image tracking can detect and track a flat image in 3D space. This is great for building content that's augmented onto business cards, posters, magazine pages, etc.

The group provides a coordinate system that has its origin at the center of the image, with positive X axis to the right, the positive Y axis towards the top and the positive Z axis coming up out of the plane of the image. The scale of the coordinate system is such that a Y value of +1 corresponds to the top of the image, and a Y value of -1 corresponds to the bottom of the image. The X axis positions of the left and right edges of the target image therefore depend on the aspect ratio of the image.

https://docs.zap.works/universal-ar/a-frame/tracking/image-tracking/

You may use the **Properties Panel** to make the following adjustments to your 3D model:

Property	Description
Content	Replace the 3D component with another 3D component you have uploaded to the Media Library.
Transforms	Edit the **Size** (Width, Height and Depth), **Position** (X/Y/Z coordinates), and **Rotation** (degrees on the X/Y/Z axes) of the 3D model. You can also **Mirror** the 3D component and change the 3D model's layer **Order** (forwards or backwards) relative to other scene components.
Actions	Create interactivity by adding an on tap **Action** to the 3D component.
Transition Effects	Add an **Enter** or **Exit** transition effect to the 3D component.

virtual object positioned within the augmented reality image based on the decoded data from the DEM and the marker location

https://docs.zap.works/designer/3d/using-3d/

https://www.youtube.com/watch?v=FY1bDxyI2Aw&t=225s